

**Informe PSIV:**

# **Color Image Processing MATLAB**

Adriana Avilés (1676127), Anna Garcia (1665817),  
Paula Franco (1674234) i David Morillo (1666540)

## Índex:

Tècniques de transformació de color .....	1
- Formulació	
- Colors complementaris	
- Correccions de to i color	
Suavitzat i nitidessa .....	2
- Color image smoothing	
Sharpening utilitzant Laplacian .....	3
Image Segmentation .....	6
Pràctica	

## TÈCNiques DE TRANSFORMACIÓ DE COLORS: Paula

Permeten **modificar l'aparença** d'una imatge al **processar les seves components de color d'un model de color específic**. Existeixen diversos models de colors bastant comuns i utilitzats en el dia a dia, com el **espai RGB, HSI i CMYK**, cadascun amb les seves característiques i aplicacions. Tal i com ha explicat l'altre grup anteriorment.

### FORMULACIÓ:

Modelem les transformacions de color per les imatges multispectrals utilitzant la següent expressió:

$$S_i = T_i(r_i) \quad i = 1, 2, \dots, n$$

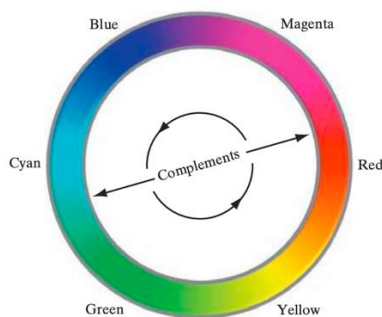
On  $n$  és el nombre total d'imatges de components,  $r_i$  són els valors d'intensitat de les imatges de components d'entrada,  $S_i$  són les intensitats corresponents espacialment a les imatges de components de sortida i  $T_i$  són un conjunt de funcions de transformació o mapeig de color que operen a  $r_i$  per produir  $S_i$ . L'equació s'aplica individualment a tots els píxels de la imatge d'entrada. Per exemple, en el cas d'imatges de color RGB,  $n=3$ ,  $r_1, r_2, r_3$  són els valors de intensitat en un punt de les imatges dels components d'entrada i  $S_1, S_2, S_3$  són els píxels transformats corresponents a la imatge de sortida. El fet que  $i$  també sigui un subíndex a  $T$  significa que, en principi, podem implementar una transformació diferent per a cada imatge de component d'entrada.

### COLORS COMPLEMENTARIS: Adriana

El **cercle cromàtic** es una representació visual dels colors que s'ordenen segons la relació cromàtica entre ells. El resultat és que els tons oposats en aquest cercle son complementaris.

Fer el complementari **recorda als negatius de les imatges**. Els vermells de la imatge original es substitueixen per blau en la imatge complementaria, i així amb tots els colors i el seu complementari respectiu. És a dir, cadascun dels tons en la imatge complementaria es poden deduir a partir de la imatge original utilitzant el cercle de color (que mostrem per pantalla – power).

En el processament d'imatges, els complements de color s'utilitzen per **ressaltar regions específiques d'una imatge**, millorant així els detalls presents en aquestes àrees.



### CORRECCIONS DE TO I COLOR: Adriana

Son tècniques utilitzades per ajustar l'aparença general d'una imatge, **corregint desequilibris tonals o variacions de color no desitjades**. Aquestes, s'apliquen a la fotografia i en la indústria del cinema per millor la qualitat i la fidelitat dels colors en les imatges.

Les **correccions de to** s'utilitzen per corregir les imatges que presenten **desequilibris en la il·luminació**, com imatges massa clares (**sobreexposades**) o massa obscures (**subexposades**).

Aquestes correccions s'ajusten als valors de les components de color d'una imatge per equilibrar el nivell de lluminositat en tota la imatge.

Les **correccions de color**, per altra banda, s'utilitzen per **corregir i millorar la reproducció dels colors** d'una imatge. Això implica ajustar els valors de les components de color per aconseguir una visió més realista o ajustar els colors a estàndards específics, com els utilitzats en la indústria de la impressió.

## **SUAVITZAT I NITIDESA EN LA IMATGE DE COLOR: Adriana**

Més enllà de transformar cada píxel d'una imatge de color com hem fet anteriorment, ara s'ha de tenir en compte als veïns.

### COLOR IMATGE SMOOTHING:

El suavitzat de les imatges en escala de grisos és una operació de filtratge en la que els coeficients del nucli de filtratge tenen el mateix valor. A mesura que el nucli es desplaça a través de la imatge que volem suavitzar, cada píxel es reemplaça amb la mitjana dels píxels del vecindari agafat.

La principal diferència en les imatges de color és que en comptes de valors d'intensitat escalar, s'ha de tractar amb vectors que contenen les 3 components de color RGB, on s'aplicarà la mitjana d'aquests vectors seguint la fórmula: (s'aplica per cada pla de color)

$$\bar{c}(x, y) = \frac{1}{K} \sum_{(s, t) \in S_{xy}} c(s, t)$$



## SHARPENING UTILITZANT LAPLACIAN: Anna

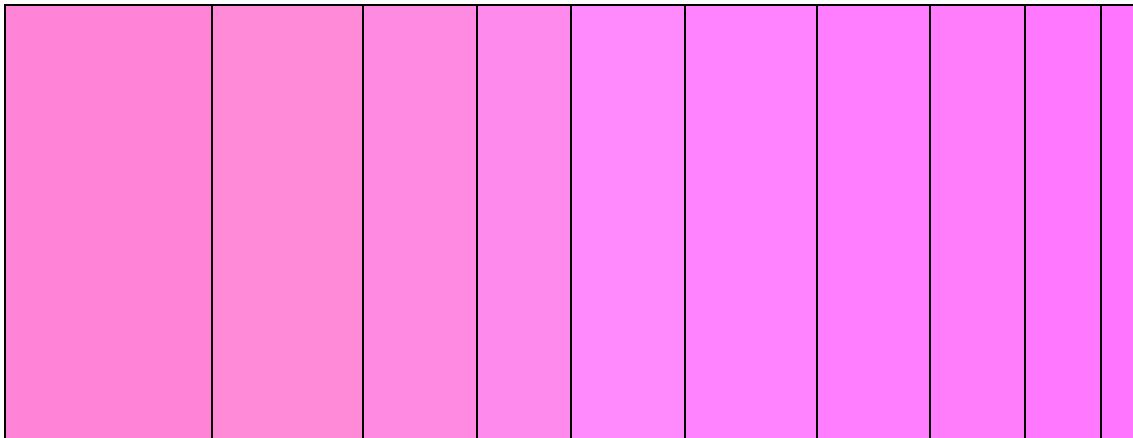
<https://www.geeksforgeeks.org/image-sharpening-using-laplacian-filter-and-high-boost-filtering-in-matlab/> -> ESTO ES EN BW 2D

Un filtre de *sharpening* s'aplica en imatges digitals per donar una **aparença més nítida**, mitjançant la **definició de les vores de la imatge**, tal i com mostra la imatge inferior:



*Imatge aplicant sharpening en diferents nivells de intensitat (0.5x i 1.5x)*

Hi ha imatges on és **difícil trobar** aquestes vores, ja que no hi ha molta diferència entre el fons i els objectes, s'anomenen **imatges sordes**. Per altra banda estan les **imatges afilades**, on els marges son clarament distingibles, ja que la intensitat i el contrast canvien just allà, contra més es noti aquest canvi en primer pla, més nítida serà la imatge.



Considerem la nitidesa de la imatge utilitzant l'operador lineal de LaPlace per cada component de color, recordem que teòricament s'aplica utilitzant la **derivada de segon ordre** degut a que es perfecte per detectar canvis bruscos o detalls en la intensitat de la imatge, que es exactament el que volem. Com les imatges són en 2D estaríem davant de les coordenades  $x$  i  $y$ , el fet d'aplicar la segona derivada respecte les coordenades ens indicaria com canvia la intensitat de la imatge en les dues direccions.

A MATLAB tenim una funció que és AQUESTA que retorna un filtre 3x3 que s'aproxima a la forma de l'operador LaPlacià bidimensional. El paràmetre  $\alpha$  és un valor de  $[0, 1]$  i bàsicament s'obté a partir d'AIXÒ.

$$\nabla^2[\mathbf{c}(x, y)] = \begin{bmatrix} \nabla^2 R(x, y) \\ \nabla^2 G(x, y) \\ \nabla^2 B(x, y) \end{bmatrix}$$

Bàsicament per detectar els marges (vores) s'utilitza una tècnica anomenada *edge sharpening* on bàsicament consisteix en seleccionar un kernel i fer la convolució entre ell (serà el filtre laplacian) i cada canal, i concatenar els resultats obtinguts, per tal d'observar-los. Tal i com mostra la imatge inferior, haureu d'obtenir-ho en la pràctica:



Els kernels que es poden implementar, ja que apliquen l'equació LaPlaciana són els següents:

0	1	0	1	1	1	0	-1	0	-1	-1	-1
1	-4	1	1	-8	1	-1	4	-1	-1	8	-1
0	1	0	1	1	1	0	-1	0	-1	-1	-1

(a) Laplacian kernel used to implement Eq. (3-53). (b) Kernel used to implement an **extension** of this equation that includes the **diagonal terms**. (c) (d) Two other Laplacian kernels: aplicant el "complementari"

Afegir el Laplacian a la imatge original restaura les variacions d'intensitat general de la imatge. L'addició del laplacian va augmentar el contrast en els llocs de discontinuïtat d'intensitat.

PAULA:

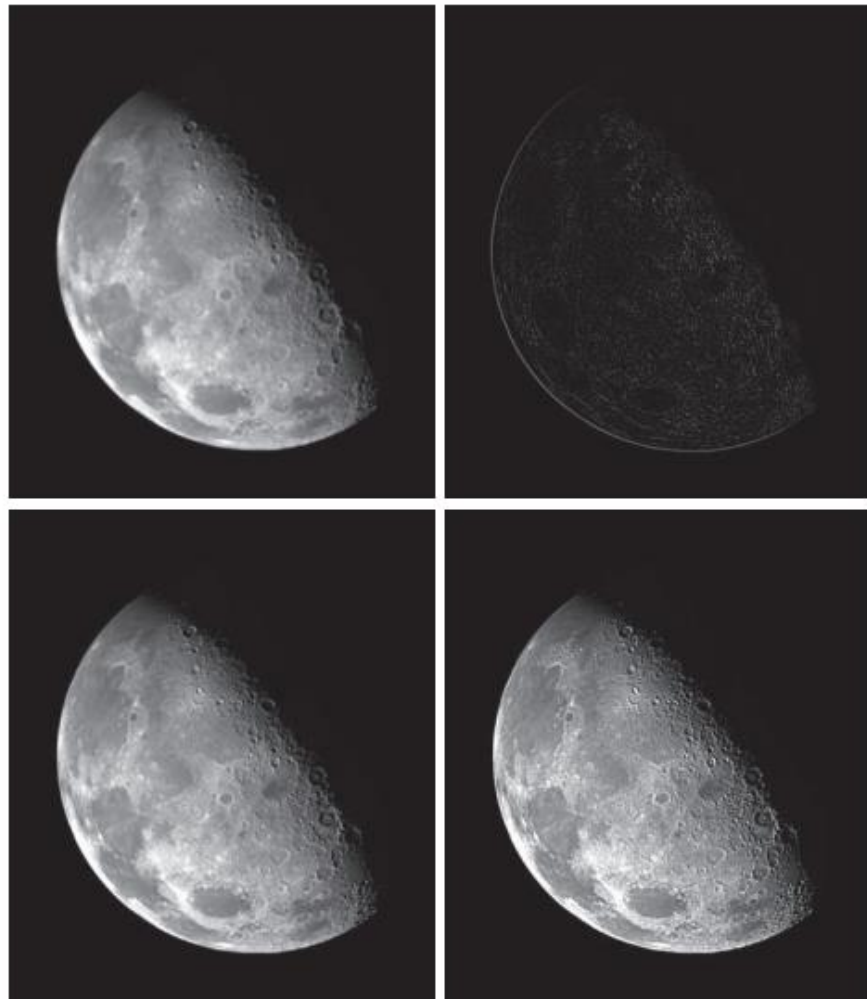
a	b
c	d

FIGURE 3.46

(a) Blurred image of the North Pole of the moon.  
 (b) Laplacian image obtained using the kernel in Fig. 3.45(a).  
 (c) Image sharpened using Eq. (3-54) with  $c = -1$ .  
 (d) Image sharpened using the same procedure, but with the kernel in Fig. 3.45(b). (Original image courtesy of NASA.)

1	1	1
1	-8	1
1	1	1

3.45(b)



Finalment, la Fig. 3.46(d) mostra el resultat de repetir el mateix procediment però utilitzant el nucli a la Fig. 3.45(b). Aquí, observem una millora significativa a la nitidesa amb respecte a la Fig. 3.46(c). Això no és inesperat perquè utilitzar el nucli en la Fig. 3.45(b) proporciona una diferenciació addicional (sharpening) a les direccions diagonals. Resultats com el de les figures 3.46(c) i (d) han fet del Laplacian una eina d'elecció per afilar imatges digitals.

FIGURE 3.47

The Laplacian image from Fig. 3.46(b), scaled to the full  $[0, 255]$  range of intensity values. Black pixels correspond to the most negative value in the unscaled Laplacian image, grays are intermediate values, and white pixels corresponds to the highest positive value.



Degut a que **les imatges laplacianes tendeixen a ser fosques i sense característiques**, una forma típica de escalar aquestes imatges per la seva reproducció és utilitzar Eqs. (2-31) i (2-32). Això porta el valor més negatiu a 0 i mostra tot el rang d'intensitats. La figura 3.47 és el resultat del processament de la figura 3.46(b) d'aquesta manera.

$$g_m = g - \min(g) \quad (2-31)$$

$$g_s = K [g_m / \max(g_m)] \quad (2-32)$$

com a resultat de l'escala. Aquesta aparença grisenca és típica de les imatges laplacianes que s'han escalat correctament.

Les característiques dominants de la imatge són les arestes i les discontinuïtats d'intensitat nítida. El fons, que abans era negre, ara és gris



## IMAGE SEGMENTATION: David

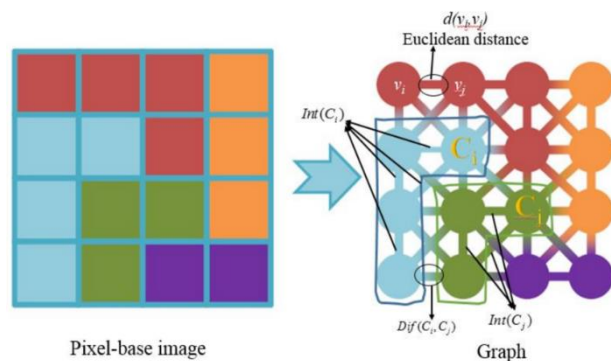
Una forma de processament d'imatges inclou la segmentació, separació o distinció d'elements formats en una imatge o vídeo. En aquesta pràctica ens enfocarem les bases d'aquesta tècnica i en una eina de MatLab anomenada Image Segmenter, inclosa en la Image Processing Toolbox, i les seves utilitats i aplicacions. Més concretament, s'explicarà la segmentació d'imatges per grafs.

Graph image segmentation:

Podem modelar qualsevol imatge digital com un graf, en el qual els nodes són cadascun dels píxels, i tenim una aresta per cada parell de nodes (o una aresta per veí per cadascun dels nodes), representant la diferència euclidiana entre els valors d'aquests píxels.

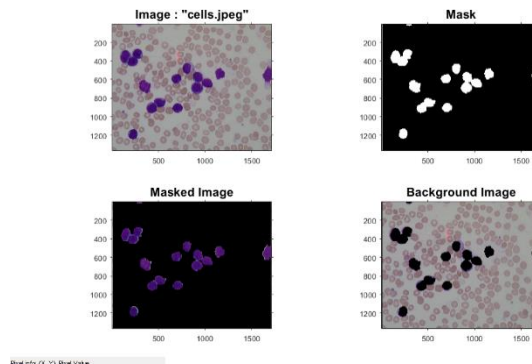
Un cop calculat aquesta mena de xarxa de píxels, l'utilitza una de les següents tècniques per a dividir el graf en diverses components:

- Spectral Clustering
- Normalized Cut
- Minimum Cut



En l'assignatura d'Anàlisi de Grafs i Cerca d'Informació es va estudiar l'algoritme de minimum cut i, per tant, no indagarem en el seu funcionament.

El que ens interessa saber d'aquest algoritme és que agrupa grups de píxels similars per cada iteració, fins a obtenir un tall d'arestes que maximitza les diferències entre píxels. Un cop separat el graf, podem modelar una màscara binària, per a combinar-la amb la imatge original i obtenir-ne dues separades, diferenciant el que ens interessa (*foreground*) del que no (*background*).



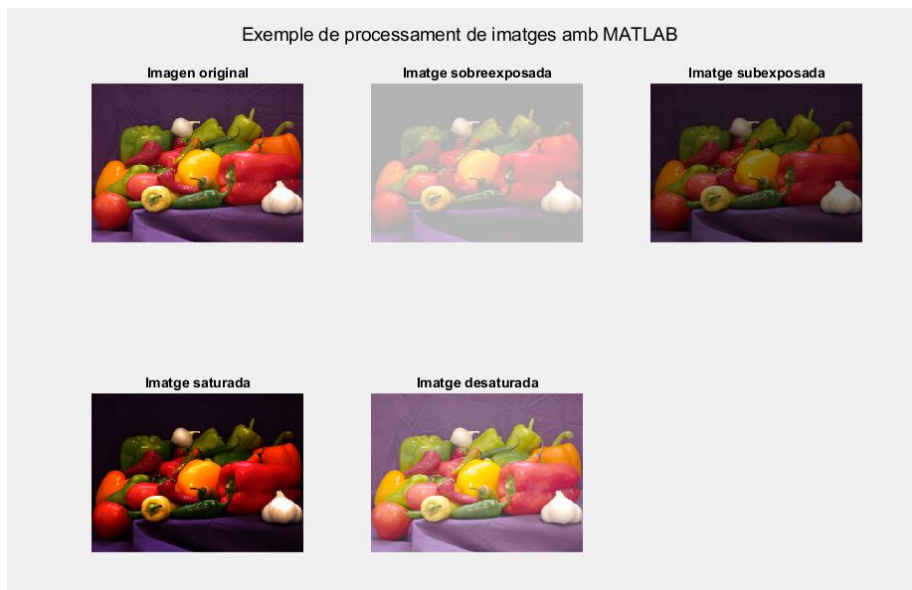
## PRÀCTICA:

**IDEAS EXPLICAR UN POCO DE LA PRACTICA LO QUE VAMOS A HACER DE CADA IDEA**

### PROCESSAMENT BÀSIC AMB MATLAB

Per començar amb el processament d'imatges de color la idea es realitzar diferents correccions de color i to a la mateixa imatge. En primer lloc, aplicarem la correcció de to per tal de simular una sobreexposició i altre per simular la subexposició. Després aplicarem dues correccions de color per mostrar la imatge amb la saturació augmentada i reduïda.



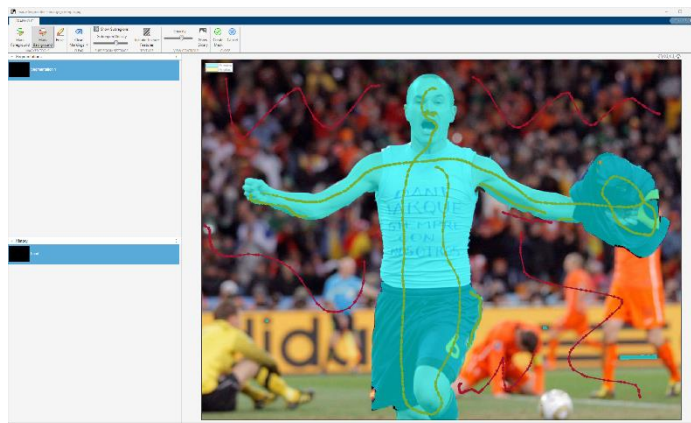


Solució

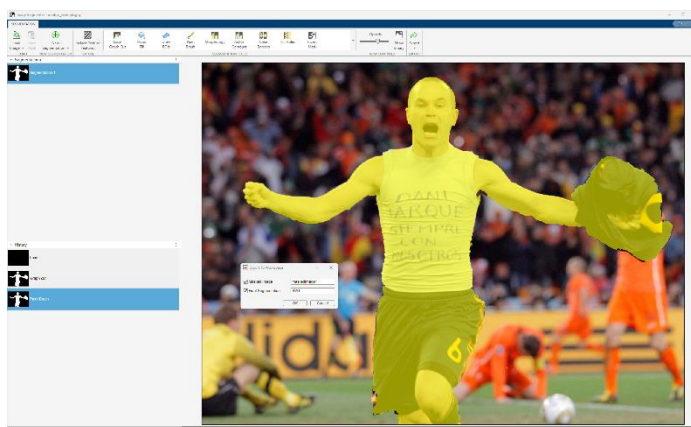
## SEGMENTACIÓ

La idea d'aquest exercici es a partir d'una imatge, retallar un objecte que ens interressi fent servir l'eina de segmentació d'imatges de MatLab, per a exportar una màscara i aplicar-la a l'imatge original.

El procés per a obtenir la màscara demandada hauria de ser semblant a aquest:



A la eina "Graph Cut" de "Image Segmentator", es selecciona el objecte que volguem, i el fons, amb els colors corresponents.



S'aplica el tall i s'exporta la màscara binària al workspace en forma de variable.

```

rgbImage = imread("imatge_exemple.jpg"); % Podeu utilitzar la imatge inclosa
mask = BW; % Variable resultant d'exportar la màscara amb la eina "Image Seg

fontSize = 14;
subplot(2, 2, 1);
imshow(rgbImage, []);
caption = sprintf('Imatge');
title(caption, 'FontSize', fontSize, 'Interpreter', 'None');

subplot(2, 2, 2);
imshow(mask, []);
title('Mask', 'FontSize', fontSize, 'Interpreter', 'None');

maskedRgbImage = bsxfun(@times, rgbImage, cast(mask, 'like', rgbImage));
subplot(2, 2, 3);
imshow(maskedRgbImage, []);
title('Masked Image', 'FontSize', fontSize, 'Interpreter', 'None');

backgroundImage = bsxfun(@times, rgbImage, cast(~mask, 'like', rgbImage));
subplot(2, 2, 4);
imshow(backgroundImage, []);
title('Background Image', 'FontSize', fontSize, 'Interpreter', 'None');

```



Finalment s'executa el codi donat i es mostra el resultat.

## SHARPENING UTILITZANT LAPLACIAN

## **Webgrafia**

<https://matlabacademy.mathworks.com/es/details/image-processing-with-matlab/mlip>

<https://es.mathworks.com/matlabcentral/answers/484801-laplacian-for-image-sharpening-implementation>

<https://es.mathworks.com/discovery/image-segmentation.html>

<https://www.baeldung.com/cs/graph-based-segmentation>

<https://youtu.be/2lVAznQwdS4>